EDRN BIOMARKER DATABASE

Engineering Plan

Dan Crichton
Sean Kelly
Jane Oh
Tina Xiao
NASA Jet Propulsion Laboratory

Don Johnsey
National Cancer Institute

1. INTRODUCTION

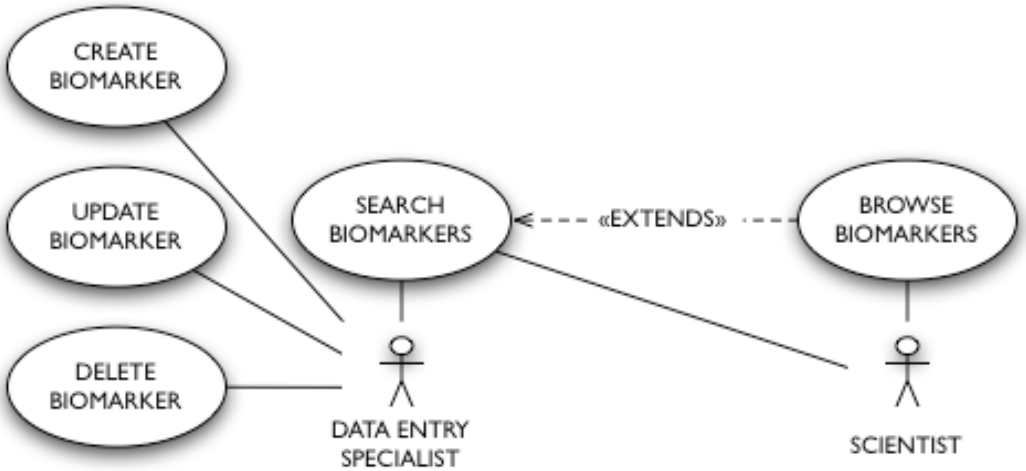
To reduce cancer mortality, the cancer must be detected sufficiently early to enable the efficacy of treatment. Indeed, early detection is the best weapon against this post potent of diseases. Recent advances in imaging technology and the emergence of novel proteomic technologies have prompted the search for *biomarkers* for the early detection of cancer. Biomarkers are biologic processes and their results that indicate the presence of or the predisposition for specific cancers.

With a *biomarker database*, researchers can develop, track, validate, coordinate, and correlate multiple cancer biomarkers. We propose to create and deploy a Biomarker Database for the Early Detection Research Network of the Division of Cancer Prevention. This document describes the engineering plan the NASA Jet Propulsion Laboratory will undertake towards that end.

2. REQUIREMENTS

This section presents the requirements of the Biomarker Database using use cases. Use cases are the preferred method of defining requirements in the Unified Software Development Process since they're user-centric (focusing on steps an actual user may perform) and testable (providing metrics for completion).

The following UML diagram depicts the use cases:



ACTORS

We define two actors who interact with the Biomarker Database:

- Data Entry Specialist

  The Data Entry Specialist may be a scientist, researcher, clinician, staff member, or other individual with the responsibility of entering and updating biomarkers into the Biomarker Database.

- Scientist

  The Scientist may be a doctor, researcher, clinician, staff member, or other individual who's interested in locating and retrieving biomarkers in the Biomarker Database.

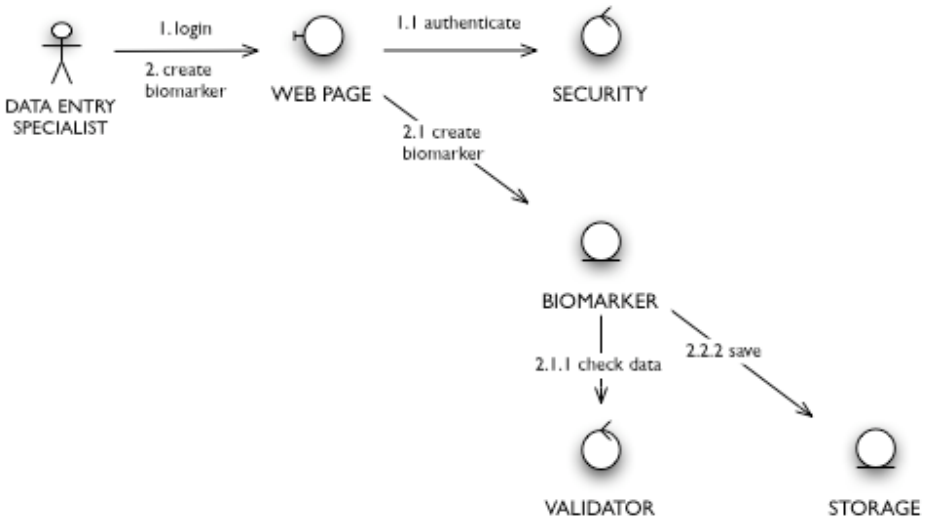
USE CASES

This section details each of the use cases depicted in the above UML diagram.

CREATE BIOMARKER

This use case begins when the Data Entry Specialist first logs onto the Biomarker Database web page. After authentication, the actor selects an action to create a new biomarker. A data entry screen appears that enables the actor to fill in details regarding the biomarker. After verifying the information, the actor presses a Save button to save changes to the database.

The architecture of this interaction appears in the following UML diagram:

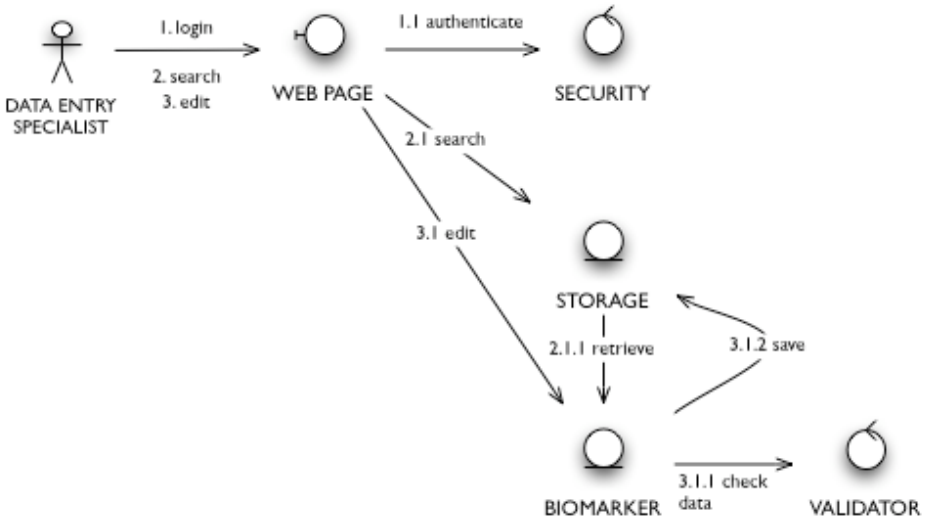


Successful termination of this use case occurs with a new biomarker saved into persistent storage. The case ends in failure if the Data Entry Specialist fails to authenticate or if the necessary details of the biomarker aren't consistent.

UPDATE BIOMARKER

This use case begins when the Data Entry Specialist logs onto the Biomarker Database web page. After authentication, the actor searches for an existing biomarker. Displaying the biomarker details, the actor selects an action to update the biomarker. Presented with an edit screen, the actor changes the details and presses a Save button to save changes to the database.

The architecture of this interaction appears in the following UML diagram:

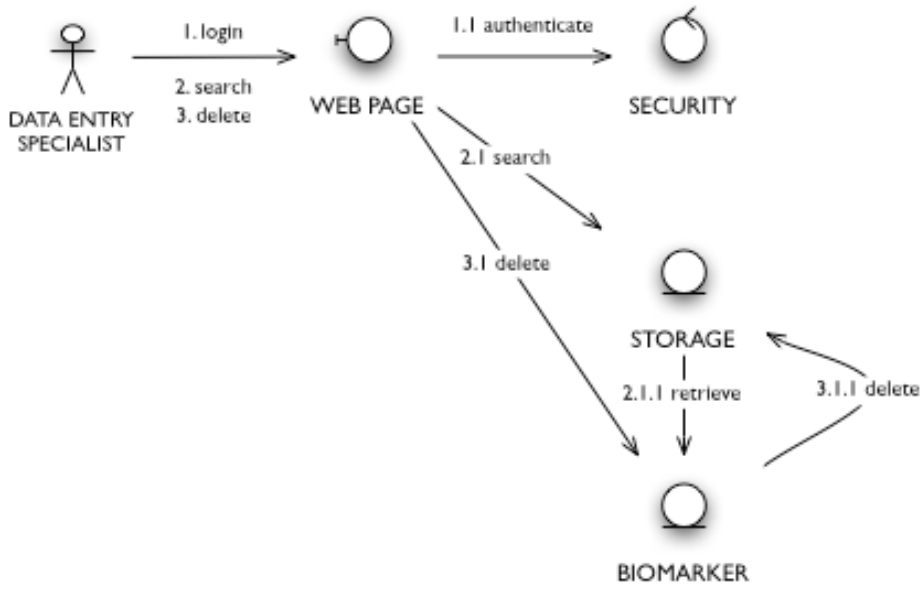


Successful termination of this use case occurs when a biomarker's details are updated and saved into persistent storage. The case ends in failure if the Data Entry Specialist fails to authenticate or if the new details of the biomarker are inconsistent.

DELETE BIOMARKER

This use case begins when the Data Entry Specialist logs onto the Biomarker Database web page. After authentication, the actor searches for an existing biomarker. Displaying the biomarker details, the actor selects an action to delete the biomakrer. Presented with a confirmation screen, the actor then confirms the action. The Biomarker Database then deletes the biomarker from persistent storage.

The architecture of this interaction appears in the following UML diagram:



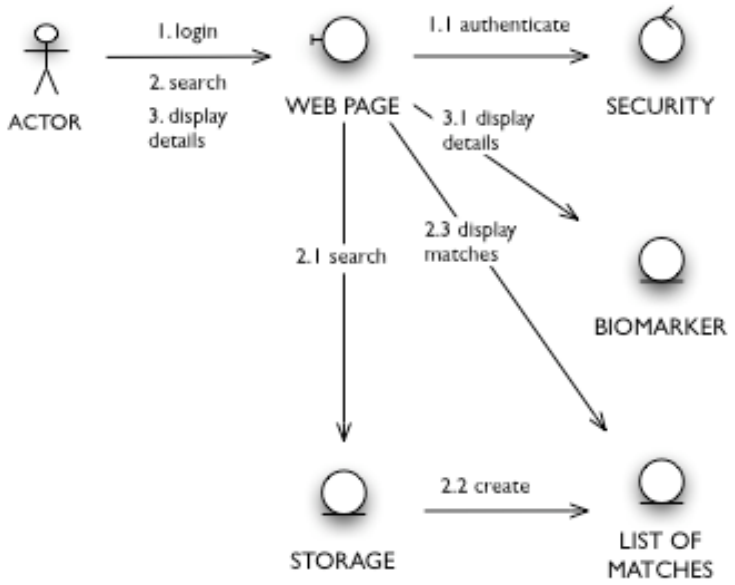
Successful termination of this use case occurs when the biomarker is deleted from persistent storage or if the deletion is not confirmed. The case ends in failure if the Data Entry Specialist fails to authenticate.

SEARCH BIOMARKERS

This use case begins when an actor logs onto the Biomarker Database web page. After authentication, the actor chooses an action to search for biomarkers. Presented with a search criteria screen, the actor enters criteria and executes the search. The web page displays summary information for all matching biomarkers (grouped into pages for large numbers of matching results). The

actor may select one biomarker at a time for viewing complete details.

The architecture for this interaction appears in the following UML diagram:



Successful termination of this use case occurs when a search is completed. The case ends in failure if the actor fails to authenticate.

BROWSE BIOMARKERS

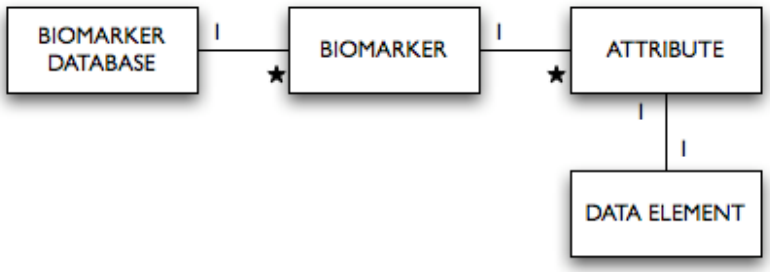
This use case begins when an actor logs onto the Biomarker Database web page. After authentication, the actor chooses an action to browse available biomarkers. The web page displays summary information for all biomarkers (grouped into pages of 10 per page) ordered by biomarker's name. The actor may select one biomarker at a time for viewing complete details.

This use case is identical to the Search Biomarkers case, except that the search is unconstrained, and all matching biomarkers are returned.

3. ARCHITECTURE

The Biomarker Database consists of a simple data model that allows for open-ended detailing of individual biomarkers. In addition, there is a user interface model that accesses the database. We present these models at a high level to validate the use cases.

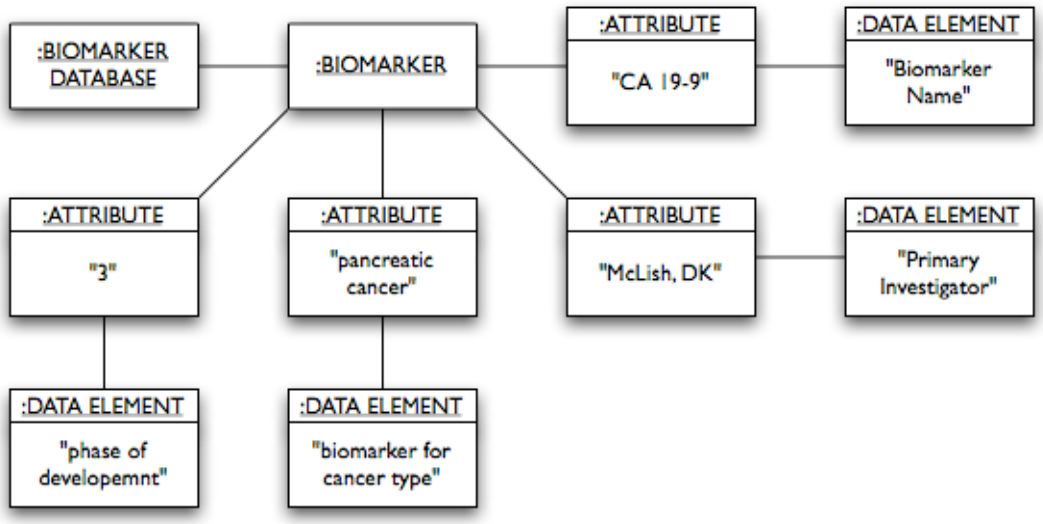
The data model is depicted in the following UML class diagram:



The Biomarker Database class is a singleton that contains all Biomarker objects. A Biomarker itself has no identifying attributes (save for internal identifiers used by the implementation). Instead, all interesting details of a Biomarker at stored in Attribute objects. Each Biomarker has zero or more Attribute objects that contain actual characterizing values of the Biomarker, such as name, ROC values, protocol, discoverer, date discovered, current phase of development, and so forth.

The definitions of these various details come from Data Element objects. Data Element objects provide the meaning and formats of legal values. Each Data Element object is defined using the EDRN Common Data Elements.

As an example, consider the following object diagram:



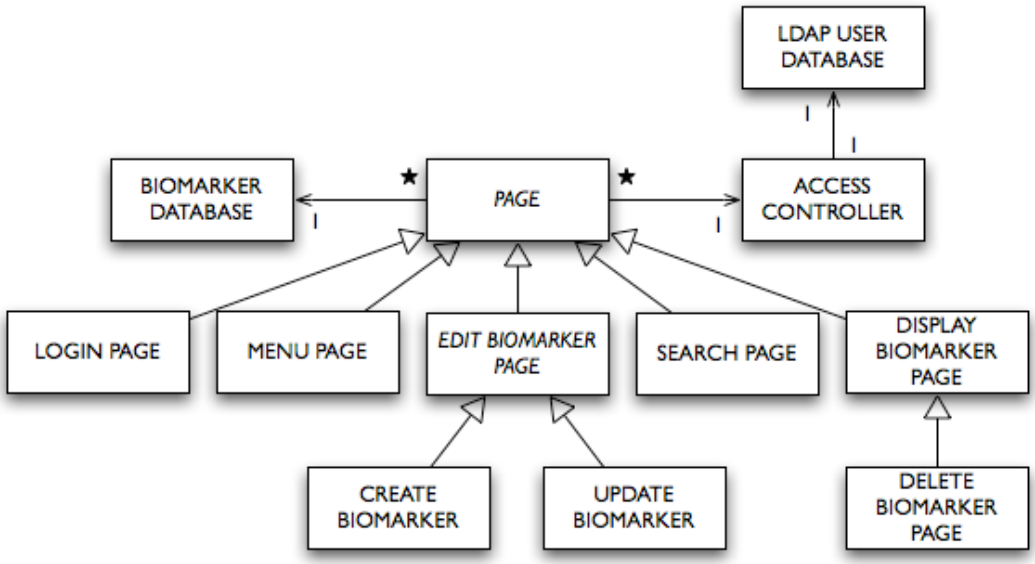
It demonstrates a single biomarker in the Biomarker Database named CA 19-9. This biomarker is currently in phase 3 of

development. It's a biomarker for pancreatic cancer, and the primary investigator is Dr. D.K. McLish.

USER INTERFACE MODEL

The classes of the user interface implement a set of web pages with access based on user identity. We expect identity to come from a forthcoming LDAP data source maintained by EDRN for all EDRN members.

The following UML class diagram shows the architecture:



An abstract Page class serves as the foundation from which all other web pages are built. Every page has access to a single Access Controller whose responsibility is to check for an authentic user. Every page also has access to the single Biomarker Database providing persistent storage for all Biomarker objects (and their corresponding Attribute objects, defined by Data Element objects).

Concrete Page subclasses implement specific parts of the user interface. The Edit Biomarker Page class is abstract, and factors out common behavior (data entry forms) used when both creating a biomarker and updating an existing biomarker.

The Delete Biomarker Page specializes the Display Biomarker Page by adding delete and delete-confirmation actions. The other page objects should be self-explanatory.

4. USER INTERFACE

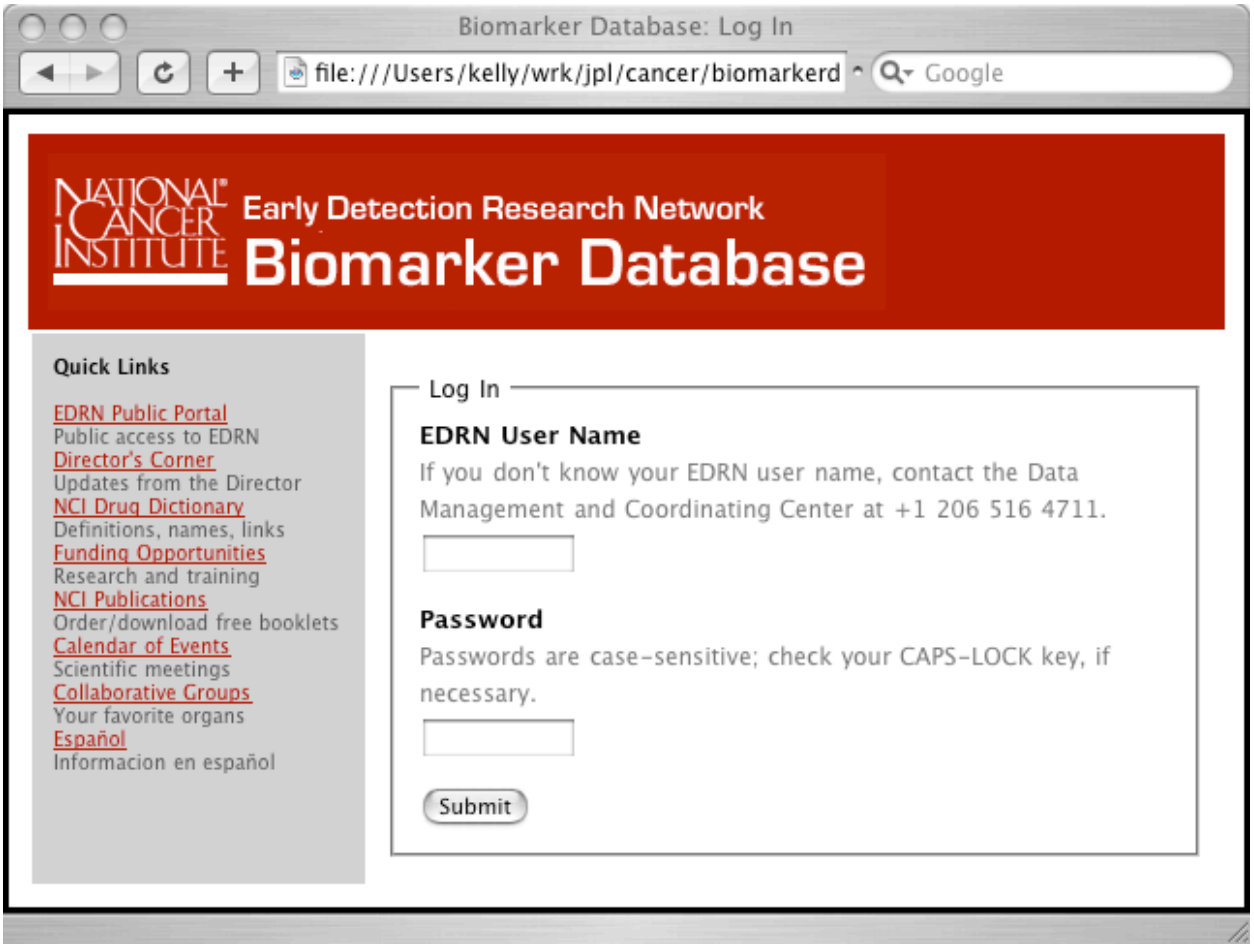
This section presents some mockups of various screens we envision existing for the Biomarker Database's user interface. These mockups are not normative; they are intended to complement the use cases and demonstrate the capabilities of the Biomarker Database.

The Biomarker Database's user interface is completely web accessible. It will not be a product that's installed onto end-user's desktop computers, but instead will be a provided within the context of a web browser. The Biomarker Database shall adhere to current web standards and we do not expect it to take advantage of advanced (and sometimes non-portable) web features such as Java, Flash, and so forth.

LOGIN SCREEN

Access to the Biomarker Database will be controlled. EDRN members will have access. Certain EDRN members (termed "Data Entry Specialists" in the use cases) will also have the ability to create and update biomarkers. Accessing the database will require the user to log in through their web browser.

The log in screen may appear as follows:

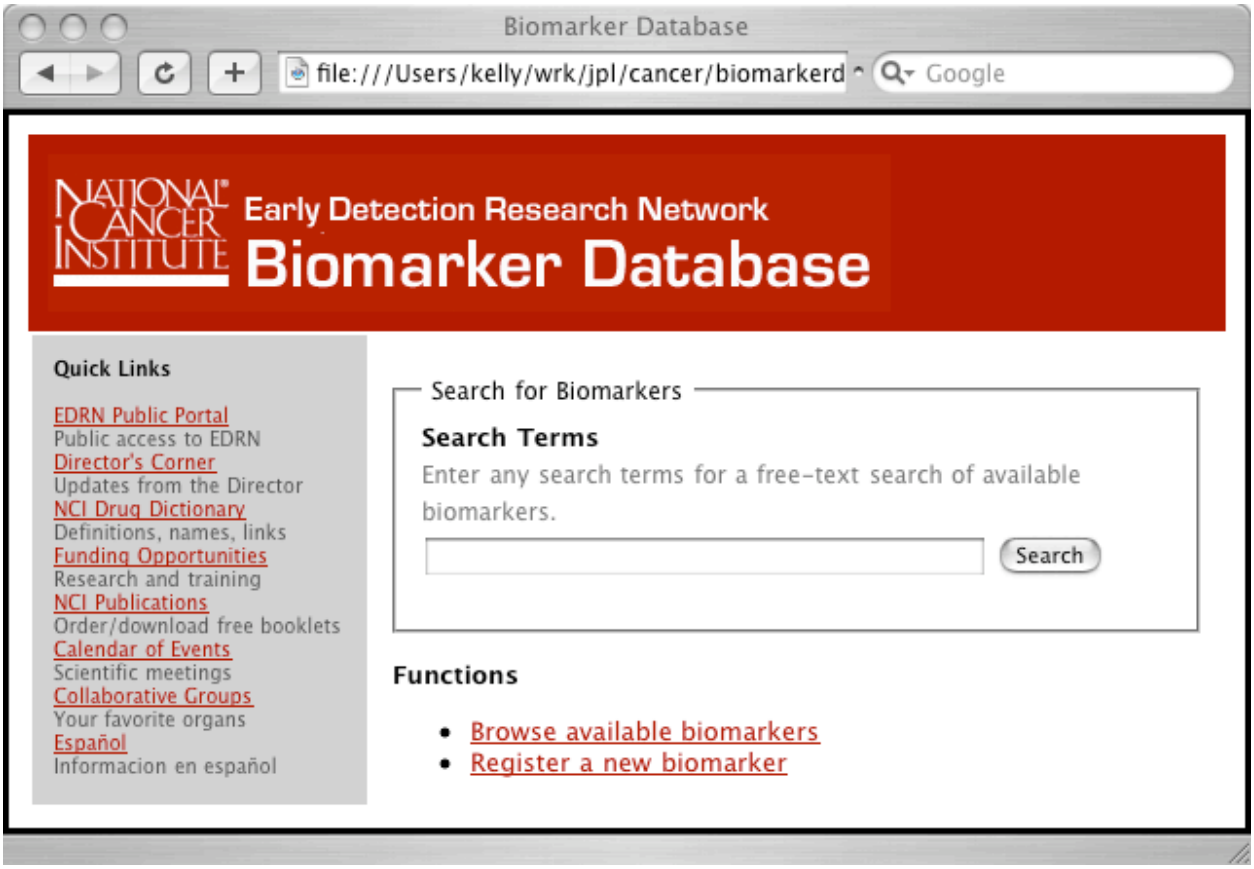


As previously mentioned, user names and passwords shall be assigned throughout the EDRN.

MAIN SCREEN

After logging in, users are greeted with the main screen, which enables searching for existing biomarkers, browsing for biomarkers, and (if users have sufficient permissions) registering a new biomarker.

The main screen appears as follows:

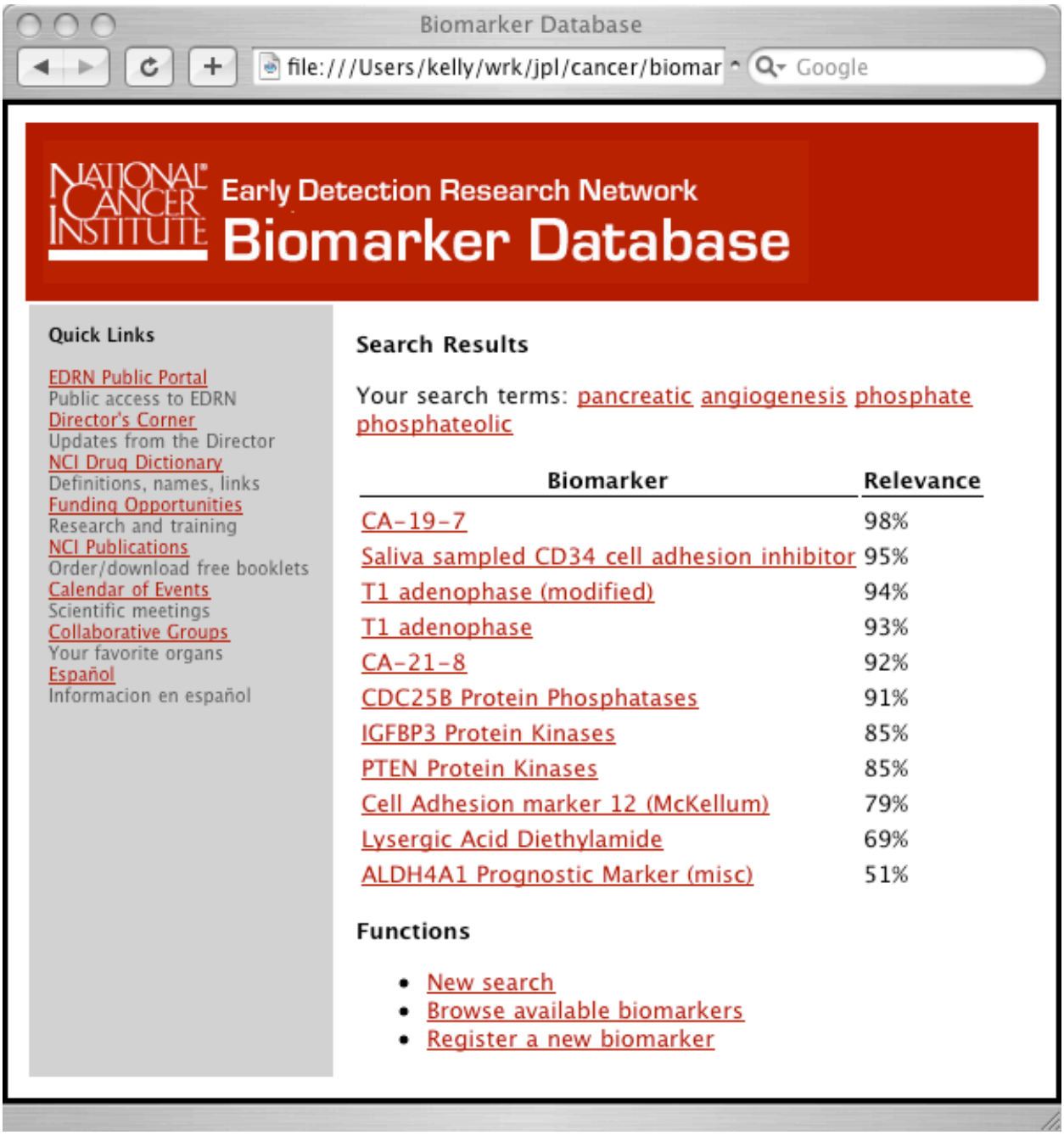


The main part of the main screen is a search area which provides a Google-like search for biomarkers. This search works across all attributes of a biomarker. Additional functions appear below the search area. Recall from above that the browsing function is equivalent to an unconstrained search, which enables the user to page through every biomarker in the database.

SEARCH RESULTS

Searching for existing biomarkers by entering some search terms results in a page that displays matching results sorted by relevance. Results are grouped in case there are many, many results. Clicking on a matching biomarker brings up additional details about it.

The screen appears like the following:

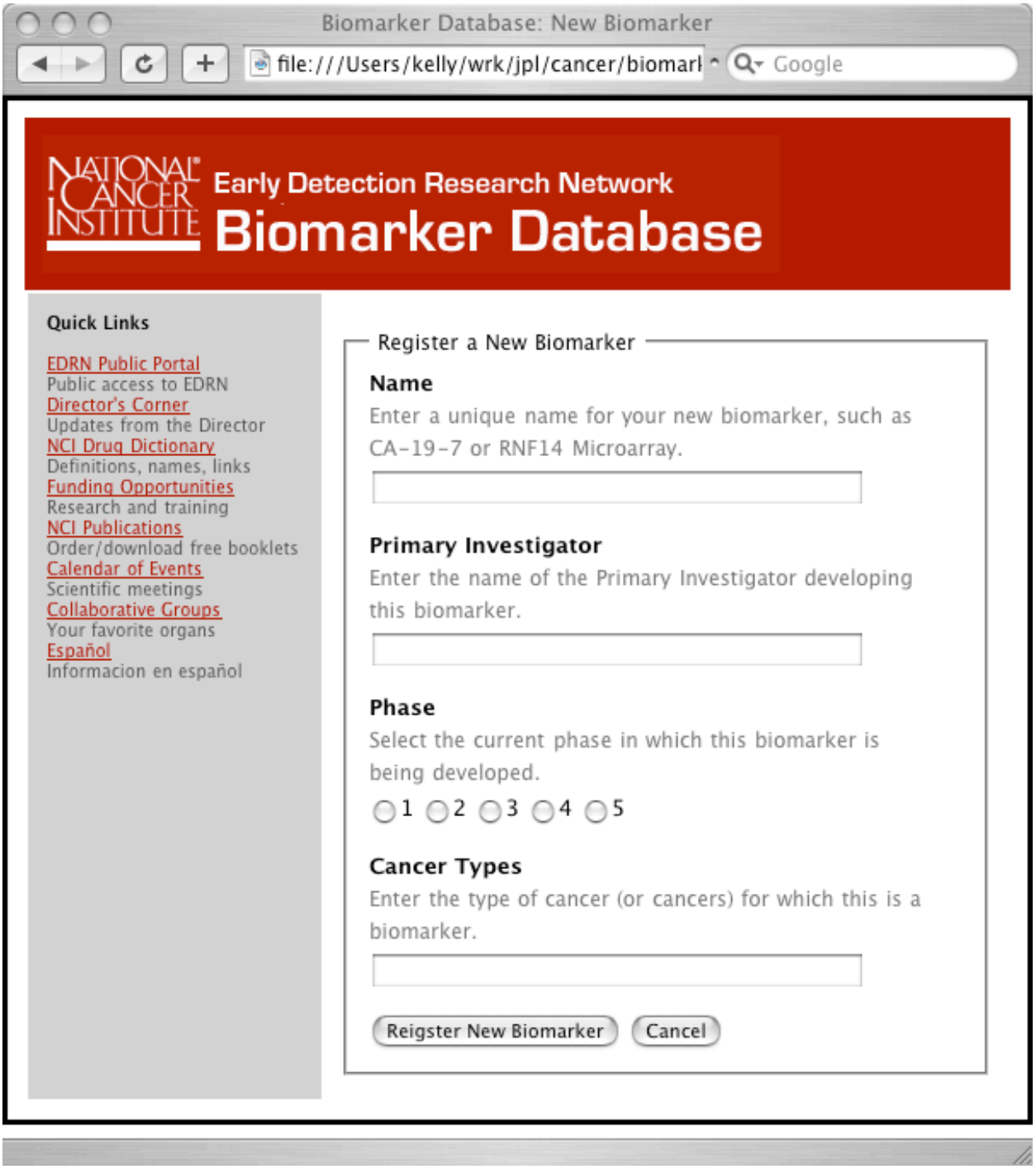


The original search terms appear as hyperlinks, which enable the user to re-execute the search without the other terms, potentially broadening the results.

The functions that appear after the search list includes the “Register a new biomarker” item, which, again, depends on the role of the user logged in.

When a user with sufficient permissions logs in, she or he may have the ability to create a new entry in the biomarker database. We term this "creating a new biomarker," but for the user interface it's termed "registering a new biomarker," since the act of physically creating it occurs outside of the context of the Biomarker Database.

Clicking the "Register a new biomarker" link presents a screen that may appear as the following:

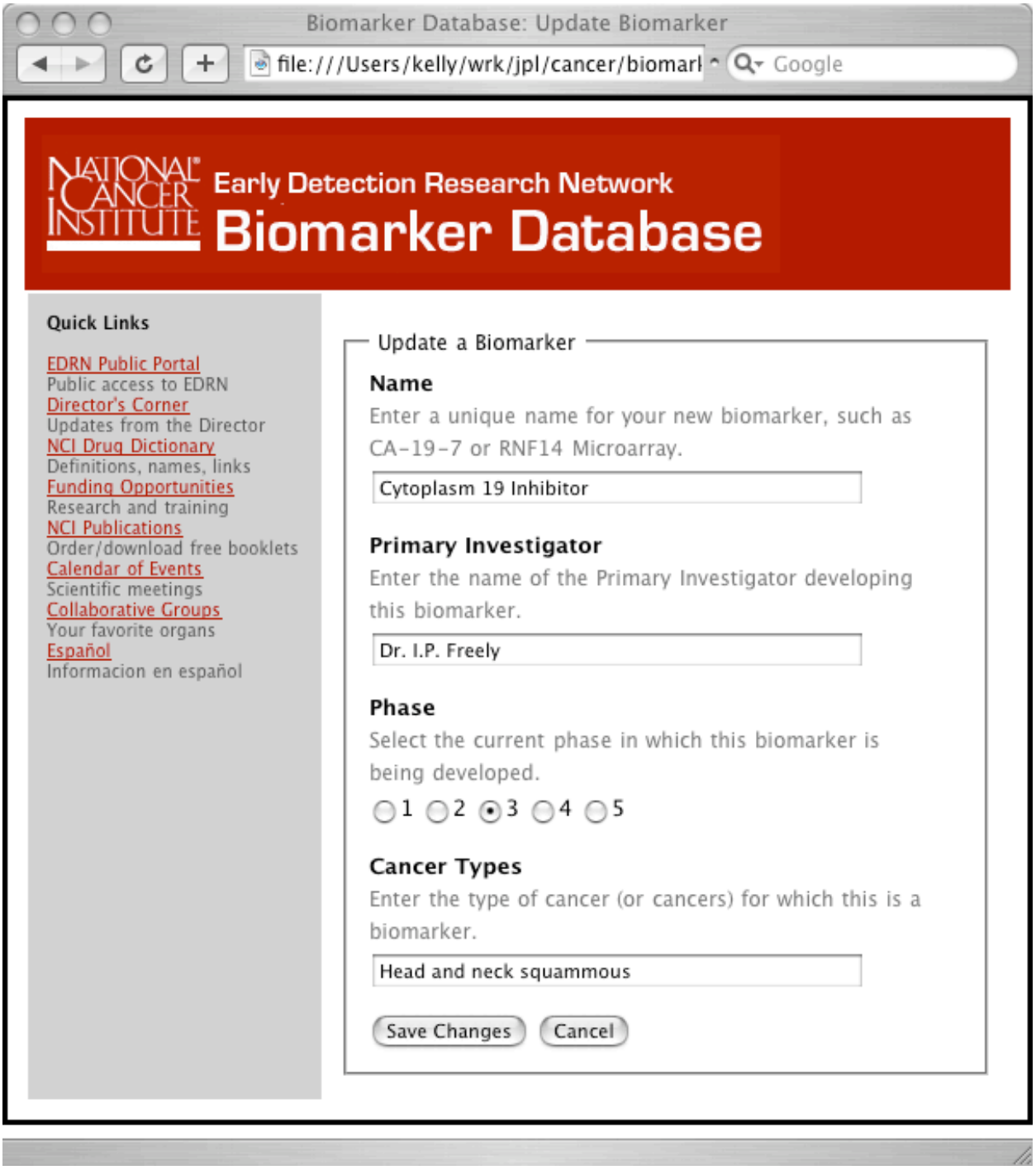


Note that the fields presented to the user to fill out on this screen are dependent on what Data Element objects exist in the database. The more Data Elements defined, the more specific the user may define the new biomarker. Further, validation of the user data is another feature that will be addressed in a future release of the Biomarker Database.

UPDATING AN EXISTING BIOMARKER

Using either the browsing capability or the search function can enable a user to select to view the details of a specific biomarker. Those users with the appropriate permissions can also edit those details once viewed.

Updating a biomarker presents a screen quite similar to the new biomarker screen, but with some labels changed, and data already filled in the various fields:



As before, the number of fields presented and their organizations may change as Data Elements mature within the Biomarker Database.

5. IMPLEMENTATION PHASES

In keeping with modern development practices (such as the Unified Software Development Process), we plan on introducing incremental releases of the Biomarker Database. The initial release will provide only the core architecture and basic features. Later releases will build on that architecture and implement additional features.

We will execute four phases: inception, elaboration, construction, and transition.

INCEPTION PHASE

The inception phase will focus on refining requirements, including refining and expanding on the use cases presented on page 2. During this phase, we'll discuss use cases with target users and create a test plan that can confirm if the resulting project satisfies the use cases.

Little to no analysis, design, implementation, or testing will occur during this phase.

ELABORATION PHASE

The elaboration phase will focus on analyzing the use cases and the creation of an analysis model that is specified by them. The analysis model will build on the entities identified by the use cases.

Also during this phase we'll elaborate on the core architecture. Some implementation will occur that prototypes the core architecture. Little to no testing will occur during this phase.

We also expect to identify risks and create mitigation plans/paths. These include:

- Using the EDRN Common Data Elements as a model for defining biomarkers
- Using a yet-to-be-built LDAP data source for users
- Creating an alternative, temporary data source for users
- Identifying alternative user interface designs that better serve users' needs

CONSTRUCTION PHASE

The construction phase will focus on implementing the Biomarker Database. During this phase we'll use the analysis model to complete the design model and implement various features.

We expect to iterate within construction phase a number of times until all features are implemented. The feature rollout will be as follows:

- Iteration 1: logging in and displaying the main screen
- Iteration 2: creating a new biomarker and defining initial minimal set of CDEs necessary to define a biomarker
- Iteration 3: browsing existing biomarkers
- Iteration 4: updating and deleting biomarkers and defining additional CDEs to refine the definition of biomarkers
- Iteration 5: searching for biomarkers, adding yet more CDEs

Within each iteration, testing will occur and test reports will show the percentage of use cases satisfied. We expect the number of use cases satisfied to rise to 100% by the end of the fifth iteration. Testing with end users will also occur, and use cases may be revised depending on feedback.

TRANSITION PHASE

The transition phase will focus on polishing any user interface elements and preparation of system documentation. Such documentation will include database administration guides, system management guides, and so forth.

During this phase, we expect to turn over management and hosting of the Biomarker Database to EDRN.

NASA Jet Propulsion Laboratory will be custodians of the system source code and will respond to bug reports filed by EDRN staff, providing patches as necessary.